

2005

Dynamic p-cycles selection in optical WDM Mesh networks

Kungmang Lo
Edith Cowan University

Daryoush Habibi
Edith Cowan University

Quoc Viet Phung
Edith Cowan University

Hoang Nguyen
Edith Cowan University

Byungkyu Kang
Edith Cowan University

Follow this and additional works at: <https://ro.ecu.edu.au/ecuworks>



Part of the [Engineering Commons](#)

[10.1109/ICON.2005.1635625](#)

This is an Author's Accepted Manuscript of: Lo, K. , Habibi, D. , Phung, Q.V. , Nguyen, H. N., & Kang, B. (2005). Dynamic p-cycles selection in optical WDM Mesh networks. Proceedings of 13th IEEE International Conference on Networks. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication. (pp. 844-849). Kuala Lumpur. IEEE. Available [here](#)

© 2005 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This Conference Proceeding is posted at Research Online.
<https://ro.ecu.edu.au/ecuworks/2917>

Dynamic p -Cycles Selection in Optical WDM Mesh Networks

Kungmeng Lo, Daryoush Habibi, *Senior Member, IEEE*, Quoc Viet Phung
Hoang Nghia Nguyen and Byungkyu Kang
SCHOOL OF ENGINEERING AND MATHEMATICS
EDITH COWAN UNIVERSITY
PERTH, AUSTRALIA

Abstract— P -cycles have been recognized as a useful protection scheme in WDM mesh networks. This is a type of shared link protection that not only retains the mesh-like capacity efficiency, but also achieves the ring-like protection switching speed. However, finding the optimal set of p -cycles for protecting traffic demands is not a simple task and is an NP-hard problem. A general approach is to determine a set of candidate p -cycles and then determine optimal or near-optimal solutions by using Integer Linear Programming (ILP) models or heuristics. In a dense mesh network, however, the number of candidate cycles is huge, and increases exponentially if the node number is increased. Thus, searching for a suitable set of efficient candidate cycles is crucial and imperative to balancing the computational time and the optimality of solutions.

In this paper, we propose a Dynamic P -cycles Selection (DPS) algorithm that improves the efficiency of enumerating candidate p -cycles. The dynamic approach for cycle selection is based on the network state. In the DPS algorithm, all cycles are found and stored, then an efficient and sufficient set of p -cycles is extracted to achieve 100% working protection, minimize the spare capacity, and reduce time complexity.

I. INTRODUCTION

The backbone traffic in optical networks is rapidly increasing due to an abundance of network services. Wavelength-division multiplexing (WDM) offers the capability to handle the increasing demand of network traffic in a manner that takes advantage of the already deployed optical fiber. Today, we have networks that carry tens of wavelengths on a single fiber, each wavelength carrying a data rate of up to 40 Gbps.

Given the high capacity transmission medium in DWDM networks, *survivability* is a very important issue, since failure of a fiber link may affect thousands of connections and cause huge data losses. The aim of survivability is to restore failed traffic flows via alternative backup routes. These backup routes may be either pre-configured, known as offline provisioning, or determined after a failure occurs, known as online provisioning. With both offline and online provisioning, the rerouting process may be performed between end nodes of effected traffic connections, referred to as *path protection*, or between two end nodes of the failed link, referred to as *link protection*. Research has shown that path protection offers better capacity utilization but link protection offers faster restoration [1].

In terms of capacity allocation strategies, survivability schemes can be classified as *dedicated protection* or *shared*

protection. In dedicated protection schemes, signals can be simultaneously transmitted on working routes and backup routes, while signals on shared protection schemes are only transmitted on working routes and switched to backup routes when a failure occurs. Thus, the dedicated schemes offer faster restoration but require much more spare capacity compared to shared protection schemes.

The concept of p -cycles as a shared span protection scheme was introduced in 1998 by Grover and Stamatelakis [2] [3] [4]. p -Cycle is about pre-configured protection cycles in a mesh network, and it merges the strengths of ring and mesh topologies; i.e. the fast recovery speed of ring restoration and the capacity efficiency of mesh protection. Moreover, p -cycles also support independent short-path routing of working demands, without constraints arising from the placement of protection structures. This method addresses the issue of speed constraints in mesh-based restoration, because the restoration time is determined individually by the two nodes of the failed span.

Ring protection mechanisms offer very fast protection switching time (some 10 ms), but the required spare-to-working resources ratio is at least 100%, and sometimes more than 200% in real networks. p -Cycles can be configured efficiently over a mesh network [5]. An individual p -cycle can restore one unit of working capacity on every on-cycle span, and restore two units of working capacity on every straddling span. A straddling span is a span whose two end-nodes are on the cycle, but the span itself is not on the cycle. Figure 1(a) illustrates a p -cycle (1-2-3-6-5-4-1), which provides a single restoration path for six span failures as shown in Figure 1(b), and two restoration paths are available for each of five straddling span failures (1-3, 2-4, 2-5, 2-6 and 3-5), as presented in Figure 1(c).

The factors that will influence the applicability of p -cycles include the nodal degree, network topology, network size, network capacity, wavelength utilization and traffic load. For example, the number of candidate p -cycles will be huge if the nodal degree is more than 4, e.g. pan-European COST239 network topology [5].

The p -cycle optimization problem can be tackled in two ways; non-joint and joint [6]. In the non-joint version, the working capacity distributed on each span is known before-

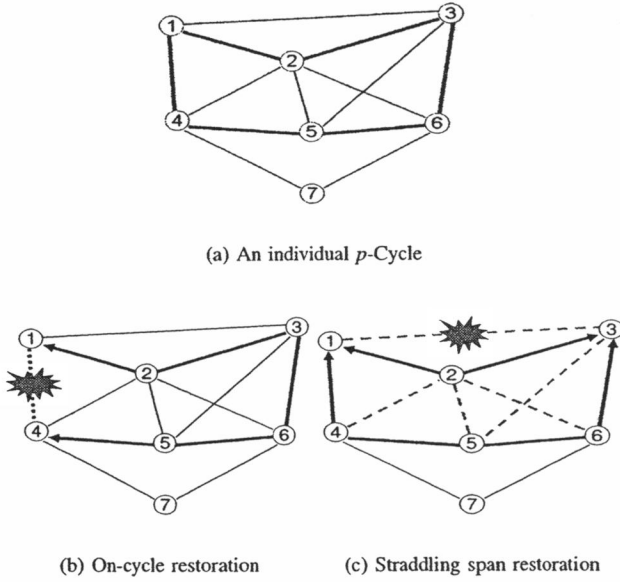


Fig. 1. The use of p -cycle for restoration

hand, and then p -cycles are generated in the spare capacity of the network [2] [7]. The objective of the non-joint version is to find an adequate set of p -cycles to protect the whole working capacity, and minimize the total spare capacity, which is used by the p -cycles. However, the routing of the demands might have to be undertaken if a protecting set of p -cycles cannot be found [8]. In the joint version, the working capacity for the traffic demands and the p -cycles are computed jointly. The objective is to minimize the sum of both capacities.

Existing approaches for solving the p -cycles problem include ILP and heuristic algorithms. Both require a set of candidate p -cycles to be pre-computed. It is usually easy to find the optimal solution for small or medium size networks because of the small number of possible cycles. For example, the network topology in Figure 1, which has 7 nodes, 13 spans, and an average nodal degree of $\bar{d} = 3.7143$, has a total of 59 possible cycles; and the NSFNet topology (14 nodes, 21 spans, $\bar{d} = 3$) has 139 possible cycles. However, the number of candidate p -cycles may be huge in large networks, posing a difficult problem to solve for both approaches [7]. For example, there are 3531 possible cycles for COST239 topology (11 nodes, 26 spans, $\bar{d} = 4.73$), and 8857 possible cycles for EON topology (19 nodes, 38 spans, $\bar{d} = 4$). Thus, finding a subset of candidate p -cycles to improve the performance is an important issue.

In this paper, we study algorithms for finding good candidate p -cycles. A Dynamic p -cycle Selection algorithm (DPS) is then proposed, which is designed to compute an efficient and sufficient subset of p -cycles. Our objective is to minimize the total amount (or cost) of the spare capacity subject to 100% p -cycle restorability. The rest of this paper is organized as follows. In section II we outline the background and review the literature. We evaluate the performance of the DPS algorithm

over arbitrary mesh topologies in section III. In section IV, we present and analyze some numerical results. Finally, the conclusion is provided in section V.

II. BACKGROUND AND RELATED WORK

In assessing the performance of a p -cycle algorithm, a predictor tool for examining its potential efficiency in protecting the working capacity is necessary. One such performance measure is *a priori efficiency*, which is defined as [9]:

$$AE(p) \triangleq \frac{\sum_{\forall i \in S} X_{p,i}}{\sum_{(\forall i \in S | X_{p,i}=1)} cost_i} \quad (1)$$

where i is a span in S , $X_{p,i}$ is the protection potential related to span i ($X_{p,i} = 1$ when span i is an on-cycle span, or, $X_{p,i} = 2$ if span i is a straddling span), and $cost_i$ is the cost of capacity on span i . If the cost of each span i is set to 1, then the total cost is the number of hops of a cycle. We refer to this case as the non-cost-weighted or logical *a priori efficiency* [7], and it can be presented as:

$$AE(p) = \frac{2 \times |S_{st,p}| + |S_{on,p}|}{|S_{on,p}|} \quad (2)$$

where $S_{st,p}$ is the number of straddling spans and $S_{on,p}$ is the number of on cycle spans. In theory, a cycle with a large AE value may have a high potential to efficiently to protect the working capacity. However, the efficiency of a cycle should also take into account the working capacity distribution.

Research into p -cycles as network protection scheme, or for network design, has generated recent interest among the research community. Zhang *et al.* propose a Cycle-based Rerouting Scheme algorithm (CRS) for link protection [10] [11]. In CRS, each link has a small set of candidate rerouting paths. They propose a Straddling Link Algorithm (SLA) to generate a small subset of p -cycles for a network graph. The generation of a cycle could be combined with the concept a pair of node-disjoint paths between the two end-nodes of a link. These two paths form a cycle, and the link is called a straddling link of the cycle. SLA is simple and fast for finding cycles, but the number of cycles may not necessarily be sufficient to achieve 100% protection.

Another scheme for finding p -cycles is presented by Doucette *et al.* [7], where they introduce a network design algorithm called CIDA (Capacitated Iterative Design Algorithm). They first generate a set of primary cycles using the SLA algorithm [10] [11], and then form complex cycles by 'Add', 'Join', or 'Grow' algorithms. CIDA is a non-joint heuristic algorithm, where one p -cycle is chosen iteratively from the set and placed in the network to reduce the unprotected working capacity until all working capacities are protected. They report that the *Grow* algorithm performs the best. Cycle efficiency is calculated by a quantity referred to as the *actual efficiency*. The actual efficiency, $E_w(p)$, is a modified version of Eq. 1, and depends not only on the number

of straddling spans and on-cycle spans, but also on the working capacities of those spans. The actual efficiency is given as

$$E_w(p) = \frac{\sum_{i \in S} w_i X_{p,i}}{\sum_{(i \in S | X_{p,i}=1)} cost_i}$$

where w_i is the amount of unprotected working capacity on span i at the time.

Liu and Ruan propose a WDCS algorithm (Weighted DFS-based Cycle Search) to find good candidate cycles for network design [6]. All cycles are generated by the Depth First Search (DFS) algorithm, and the core idea of their algorithm is to generate a small set of cycles containing of high efficiency cycles and short cycles, so that both densely distributed and sparsely distributed working capacities can be efficiently protected by those candidate cycles. To achieve their objective, each link in the network is protected by at least one high efficiency cycle and two short cycles (one is to be an on-cycle link, the other is to be a straddling link). Their results show that a small number of cycles can be efficiently chosen, and achieve near optimal solutions.

III. THE DYNAMIC P-CYCLE SELECTION ALGORITHM

A. Assumptions and Notations

As described earlier, finding an optimal set of p -cycles for protecting traffic demands is not a simple task, and it is an NP-hard problem. In this section, we propose a Dynamic P -cycles Selection algorithm (DPS) to generate a suitable set of efficient candidate cycles in order to balance the computational time and the optimality of solutions. The following factors are given due consideration in the algorithm:

- 1) The network scalability problem;
- 2) The network topology and working capacity distribution;
- 3) Efficiency and simplicity in finding an adequate set of cycles.

Our analytical model is developed under the following assumptions:

- The mesh network is a set of nodes interconnected by single-fiber links, and each fiber link is bidirectional.
- Each node has the same array of transmitters and receivers.
- Each span has the same weight. In this paper, we set the weight to one.
- There is no multi-casting, i.e. every call request is end-to-end.
- Only single link failures are considered.
- Full wavelength conversion capability is available at every nodes of the network.
- There is no node-bridge, or link-bridge in the network, i.e. the network graph G is two-connected.
- The total traffic request is known in-advance. Routing each path is done by an ILP model to minimize the total working capacity.

At this point we shall outline the notations which will be used to describe the DSP algorithm. A physical topology is modeled as a connected graph $G(N, L)$, where N is the set of network

wavelength routing nodes and L is the set of network single-fibers. The parameters of the model are:

N	the set of network nodes (e.g OXCs or OADMs).
N_i	a set of nodes, which nodal degree is larger than 2, $N_i \subset N$.
s	a source node.
n_s	the neighbour node of s .
$L = \{l_i\}$	the set of optical fiber links in the network, where $i \in \{1 \dots L\}$.
C	the total number of capacity in a fiber link.
K	the number of candidate routes for each connections.
$h_i = K_{sd}^i $	of hop numbers, $i \in \{1 \dots k\}$; $s, d \in \{1 \dots N\}, s \neq d$.
$deg(n_i)$	the hop number at the path K_{sd}^i .
On_cycle_l	the nodal degree at node n_i . In a random mesh topology, $2 \leq deg(n_i) \leq N - 1$.
$AEcycles_l$	a set of small cycles of link l that does not contain any straddling link in it. The AE value is 1.
w_j	a set of efficient cycles of link l containing at least one straddling link in it. The AE value is more than 1.
$Path_l^i$	the working capacity on the link j .
\bar{e}	the i_{th} shortest path generated from link l .
δ	the average value in a given series numbers e .
T	the standard deviation in a given series numbers.
$p_{d,k}$	the set of given traffic demands in which $t_d, d = [1 \dots D]$
$\beta_{d,k}$	is the volume of demand d and D is the number of traffic demand
	the k^{th} candidate route of connection d
	a indicator variable for candidate route $p_{d,k}$

B. The DPS Algorithm

The DPS algorithm is a two-step process which first determines all possible cycles in the network, and then selects a set of suitable candidate cycles. These two steps are described below.

Step 1: Pre-compute Span-based Cycles (PSC) algorithm

The purpose of the PSC algorithm is to compute all possible cycles for each span, and classifying and storing those cycles as oncycles or AEcycles. An oncycle is a small cycle which contains the span itself but does not contain any straddling link ($S_{st,p} = 0, AE(p) = 1$). All oncycles are organized in ascending order of the hop length. For the set of AEcycles, each cycle must contain at least one straddling link ($S_{st,p} \geq 1, AE(p) \geq 1$), that is the span itself. In addition, a set of AEcycles is arranged in a descending order of the value of *a priori efficiency* given in Eq. 2. It is worth noting that AEcycles can only be generated in spans whose end nodes are in N_i . The pseudocode of DPS algorithm is presented in Algorithm 1.

Step 2: Efficient Cycles Selection (ECS) algorithm

Given the traffic request, the ECS algorithm will first monitor the working capacity distribution to prevent undue traffic congestion occurring at any link, which could deny enough spare capacity for p -cycles. The traffic distribution is optimized by an ILP solver. The ILP model normally determines the working capacity on each span based on the shortest-path routing algorithm. This may lead to high working capacity on some spans, making it harder to find a solution for p -cycle protection. Therefore, the routing of the demands has to be adjusted if a protecting set of p -cycles cannot be found. This situation can particularly arise in those spans with a nodal

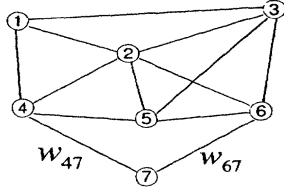
Algorithm 1 : Pre-compute span-based cycles (PSC)**Require:** $Path_i^j, \forall i \in \{1 \dots K\}, \forall j = [1 \dots |L|]$ **Ensure:** On_cycles and $AEcycles$ $cycle \leftarrow \emptyset; On_cycles \leftarrow \emptyset; AEcycles \leftarrow \emptyset$ **for** $l = 1$ to $|L|$ **do****for** $i = 1$ to $K - 1$ **do****for** $j = i + 1$ to K **do****if** $Path_i^j$ and $Path_j^i$ are node-disjoint paths **then** $cycle \leftarrow Path_i^j \cup Path_j^i$ **if** $cycle$ is an oncycle **then** $On_cycles \leftarrow (On_cycles \cup cycle)$ **else** $AEcycles \leftarrow (AEcycles \cup cycle)$ **end if****end if****end for****end for**Re-arrange the order of On_cycles_{s,n_s} in ascending cycle hop numberRe-arrange the order of $AEcycles_{s,n_s}$ in descending AE value

Fig. 2. The 7-node, 13-link topology with specific working capacities assigned to links 4-7 and 6-7

degree less or equal to two. For example, in Figure 2 consider node 7 which has a nodal degree of two. If the working capacity of links 4-7 or 6-7, given by w_{47} and w_{67} respectively, is more than half of the span capacity ($C/2$), then it is not possible to find the optimal solution for p -cycle protection. We propose an ILP model to resolve the above problem in provisioning the working routes. The objective of our ILP model is:

$$\min \sum_{j=1}^{|L|} w_j \quad (3)$$

subject to:

$$\sum_{k=1}^K \beta_{d,k} = t_d, \forall d \in T \quad (4)$$

$$w_j = \sum_{d=1}^{|D|} \sum_{k=1}^K b_{d,k}^j \times \beta_{d,k}, \forall j \in L \quad (5)$$

where

$$b_{d,k}^j = \begin{cases} 1 & \text{if } p_{k,d} \text{ uses link } j \\ 0 & \text{otherwise} \end{cases}$$

$$w_j \leq \frac{C}{2}, \text{ if } \{\forall j \in L | \text{end node of link } j \text{ is in } N \setminus N_i\} \quad (6)$$

$$w_j \leq C, \text{ otherwise} \quad (7)$$

The main purpose of ECS is to select appropriate candidate cycles according to the network state, assuming that the problem of over-allocation of working channels on links has been resolved. Each link l in the network will be assigned a value, referred to as the *span-weight*, denoted by sw_l . Let \bar{e} and δ be the average and the standard deviation of the working traffic load respectively. The span-weight is calculated based on the current working traffic load w_l :

$$sw_l = \begin{cases} 0.7 & \text{If } w_l \geq \bar{e} + 1.5\delta \\ 0.5 & \text{If } \bar{e} + \delta \leq w_l < \bar{e} + 1.5\delta \\ 0.3 & \text{If } \bar{e} \leq w_l < \bar{e} + \delta \\ 0.1 & \text{If } \bar{e} - \delta \leq w_l < \bar{e} \\ 0 & \text{If } w_l < \bar{e} - \delta \end{cases} \quad (8)$$

For each link e_l , the ECS selects a number of cycles as suitable candidates. This selection is according to the following constraints:

- If an end node of e_l has the highest nodal degree in the network, then the AE_i score of the selected i^{th} cycle on $AEcycles$ must satisfy the following condition:

$$AE_i \geq |AEcycles_l| \times (sw_l + 0.3) \quad (9)$$

- If the end nodes of e_l have nodal degrees larger than 2 and less than the highest nodal degree in the network, then the AE_i score of the selected $AEcycles$ must satisfy the following condition:

$$AE_i \geq |AEcycles_l| \times sw_l \quad (10)$$

- If link e_l is incident on a node of degree 2, then all oncycles of this link are selected.

The pseudocode of the ECS algorithm is given in Algorithm 2.

IV. NUMERICAL RESULTS AND ANALYSIS

A. Finding all possible p -cycles by the PSC Algorithm

In this section, we present the results generated by the DPS algorithm. First, we perform the PSC algorithm to find all possible p -cycles of the following three network topologies: the 7-node 13-link topology, as shown in Figure 1; the NSFNet topology; and the COST239 topology. The computing platform is an IBM ThinkCentre PC, with an Intel Pentium IV 3.0-GHz processor, with 1 GB of RAM, running Windows XP. Table I presents the performance results, given as the number of total p -cycles, the average *a priori efficiency* value for those cycles, the average cycle length, the running time, and the highest value of K used for generating the set of K shortest paths for each span's end nodes. Note that initially as we increase the value of K , the number of cycles found increases, but increasing K beyond some value (referred in Table I as the *highest value of K*) will not yield any more cycles. Therefore, in Table I, the maximum number of cycles found

Algorithm 2 : Efficient Cycles Selection (ECS)**Require:** On_cycles and AEcycles \forall span_{*l*}**Ensure:** *E*-cycles**Task 1:** Exam working capacity at each span:**for** $j = 1$ to $|L|$ **do** **if** (end nodes of link j is in $N \setminus N_i$ and $w_j > C/2$) **then**
 Re-do traffic routing by ILP from Eqs. 3, 4, 5, 6, 7 **end if****end for****Task 2:** Find *E*-cycles:**for** $j = 1$ to $|L|$ **do** $(m, n) \leftarrow$ end nodes of link j $maxdeg \leftarrow$ maximum nodal degree **if** $max(deg(m), deg(n)) = maxdeg$ **then** **if** The AE_i of cycle i satisfies Eqs. 8, 9 **then** $E\text{-cycles} \leftarrow AE\text{cycle}_j^i$ **end if** **else** **if** The AE_i of cycle i satisfies Eqs. 8, 10 **then** $E\text{-cycles} \leftarrow AE\text{cycle}_j^i$ **end if** **end if** **if** $\min(deg(m), deg(n))=2$ **then** $E\text{-cycles} \leftarrow \text{On-cycle}_j^{all}$ **end if****end for**Remove the repeated cycles in *E*-cycles

corresponds to the *highest value of K*. The results indicate that the PSC algorithm can efficiently find all possible cycles within a reasonable time. K is the number of shortest paths for the span's end nodes. Figures 3 and 4 present the the number of p -cycles, the average AE value, the average of cycle length, and the computational time, as a function of K , for NSFNet and COST239 networks.

It is worth noting that the COST239 network has a higher average AE value (2.806) and a lower average cycle length (8.748) compared to the NSFNet, for which these are 1.416 and 9.590 respectively. This means that the redundancy for COST239 network will be lower than NSFNet.

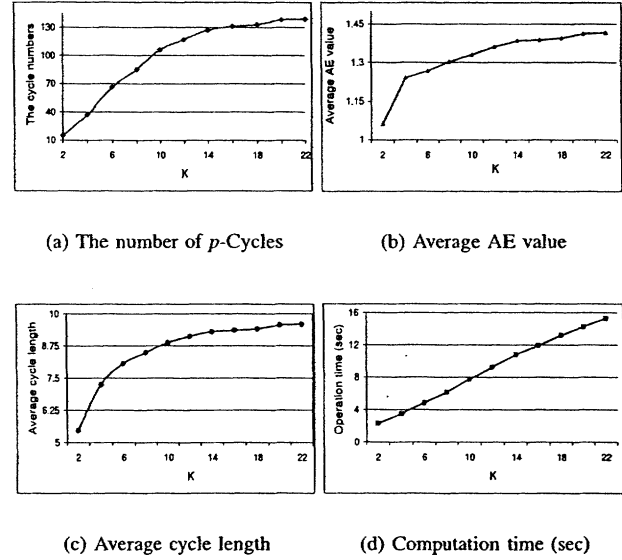
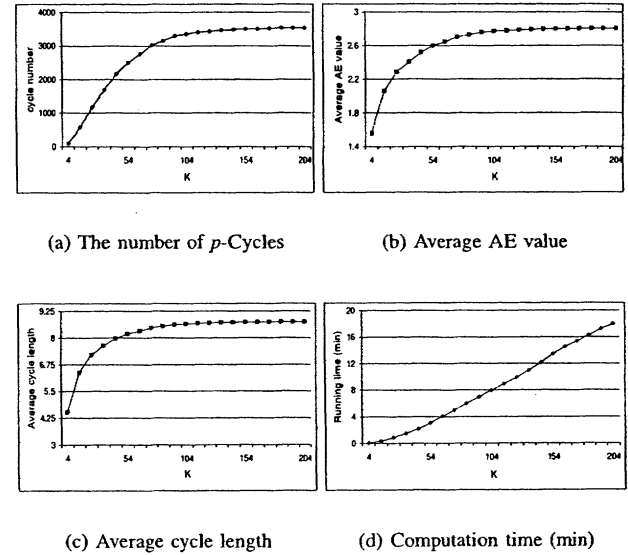
TABLE I

THE PERFORMANCE OF THE PSC ALGORITHM FOR VARIOUS TOPOLOGIES

Network topology	7-node,13-link	NSFNet	COST239
Max. no. of cycles found	59	139	3531
Average AE value	1.916	1.416	2.806
Average cycle length	5.068	9.590	8.748
Computation time (sec)	2.593	15.562	1097.8
Highest value of K	10	22	204

B. Cycle selection and performance by ECS Algorithm

In the last section we used the PSC algorithm to generate all possible p -cycles and classified them as AEcycles or On-cycles for each link. In this section we use the ECS algorithm for dynamic selection of an efficient and sufficient set of p -cycles

Fig. 3. Computing p -cycles for the NSFNet topologyFig. 4. Computing p -cycles for the COST239 topology

to achieve 100% working capacity protection and minimize the spare capacity. We then evaluate the quality of the outcome as compared to the case where all cycles are selected. In this study, we assume that a link contains 16 spans ($C=16$) and that the cost for each span is one.

Table II shows the performance of our proposed technique for the 7-node 13-link topology. The working capacity is the sum of capacity units used on all working spans. The p -cycles capacity cost is the total capacity utilized by the p -cycles. The utilization of the cycles is also presented. We find that a small number of cycles are selected for both the DPS scenario

and the select-all scenario for this topology. The simulation results show that for smaller networks the DPS algorithm gives optimal protection solution, the same as the select-all scenario. We observe that on average, the DPS algorithm has removed nearly one third of the total possible cycles, and has reduced the computation time by about 23%.

TABLE II

PERFORMANCE RESULTS FOR THE 7-NODE,13-LINK TOPOLOGY

Network	7-node,13-link (C = 16)					
Working capacity	67	68	69	70	71	72
p-cycles capacity cost	36	44	40	43	35	37
No. of candidate cycles	41	45	38	42	38	33
No. of cycles used	6	6	5	4	4	6
No. of AEcycles used	5	4	4	3	3	5
No. of oncycles used	1	2	1	1	1	1
Redundancy (%)	55.22	64.71	57.97	61.43	49.3	51.39
Computation time (sec)	10.17	9.23	8.91	7.14	6.94	10.16

Table III shows the performance results for the NSFNet topology. The observations are very similar to the last topology. In this case, the DSP algorithm has removed about 37% of total possible cycles, and has saved around 39% of the computation time.

TABLE III

PERFORMANCE RESULTS FOR THE NSFNET TOPOLOGY

Network	NSFNet (N = 14, L = 21, C = 16)				
Working capacity	112	116	117	119	123
p-cycles capacity cost	87	87	93	89	93
No. of candidate cycles	94	80	98	80	90
No. of cycles used	7	6	8	5	7
No. of AEcycles used	5	5	8	4	6
No. of oncycles used	2	1	0	1	1
Redundancy (%)	77.68	75	79.49	74.8	75.6
Computation time (sec)	31.33	74.19	48.38	43.56	43.19

Table IV shows the performance results for the COST239 network. Note that no oncycles have been selected for the protection of this topology. The same observation applies to the select-all scenario, although we have not listed the results due to the limited space. The reason is that this network does not have any node with a nodal degree less than or equal to 2. The results show that for this topology, the DPS algorithm has removed nearly 84% of the total cycles and has significantly reduced the computation time by about 88%. However, in general, the DPS algorithm can only achieve near optimal solutions for this topology.

TABLE IV

PERFORMANCE RESULTS FOR THE COST239 TOPOLOGY

Network	COST239 (N = 11, L = 26, C = 16)				
Working capacity	115	117	118	120	121
p-cycles capacity cost	49	48	51	53	51
No. of candidate cycles	540	665	571	532	570
No. of cycles used	6	6	6	7	6
No. of AEcycles used	6	6	6	7	6
No. of oncycles used	0	0	0	0	0
Redundancy (%)	42.15	41.03	43.22	44.17	42.15
Computation time (min)	7.4	102.9	15.8	32.4	43.2

V. CONCLUSION

In this paper, we have proposed the DPS algorithm for generating an efficient and sufficient set of p-cycles to achieve 100% working capacity protection, whilst also minimizing the spare capacity and reducing the time complexity. The simulation results showed that the proposed algorithm can achieve optimal solutions for medium size networks, and near optimal solutions for dense networks. We found that through our algorithm the time complexity is greatly reduced for dense networks. Thus, we can conclude that the DPS algorithm balances the computational time and the optimality of solutions, and can be applied to large networks.

REFERENCES

- [1] S. Ramamurthy, L. Sahasrabudhe, and B. Mukherjee, "Survivable WDM mesh network," *Journal of Lightwave Technology*, vol. 21, no. 4, pp. 870-883, 2003.
- [2] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed pre-configuration: Ringlike speed with mesh-like capacity for self-planning network restoration," *Proc. IEEE International Conference on Communications (ICC) 98, Atlanta, Georgia, USA*, pp. 537-543, 1998.
- [3] W. D. Grover and D. Stamatelakis, "Self-organizing closed path configuration of restoration capacity in broadband mesh transport networks," *slides presented at Second Canadian Conference on Broadband Research (CCBR 98), Ottawa, Ontario, Canada*, 1998.
- [4] W. D. Grover and D. Stamatelakis, "Bridging thenring-mesh dichotomy with p-cycles," *Proceedings of Design of Reliable Communication Networks (DRCN 2000)*, 2000.
- [5] F. J. Blouin, A. Sack, W. D. Grover, and H. Nasrallah, "Benefits of p-cycles in a mixed protection and restoration approach," *Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), Banff, Alberta, Canada*, pp. 203-211, 2003.
- [6] C. Liu and L. Ruan, "Finding good candidate cycles for efficient p-cycle network design," *Proc. 13th International Conference on Computer Communications and Networks (ICCCN 2004)*, pp. 321-326, 2004.
- [7] J. Doucette, D. He, W. D. Grover, and O. Yang, "Algorithmic approaches for efficient enumeration of candidate p-cycles and capacitated p-cycle network design," *Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003)*, pp. 212-220, 2003.
- [8] D. Schupke, C. Gruber, and A. Autenrieth, "Optimal configuration of p-cycles in wdm networks," *IEEE*, pp. 2761-2765, 2002.
- [9] W. D. Grover and J. Doucette, "Advances in optical network design with p-cycles: Joint optimization and pre-selection of candidate p-cycles," *Proc. IEEE LEOS Summer Topicals 2002, Mont Tremblant, Quebec, Canada*, pp. 49-50, 2002.
- [10] H. Zhang and O. Yang, "Finding protection cycles in dwdm networks," *Proc. IEEE International Conference on Communications (ICC)*, pp. 2756-2760, 2002.
- [11] H. Zhang, O. Yang, JingWu, J. M. Savoie, T. Shan, and G. Wang, "A cycle-based rerouting scheme for wavelength-routed wdm networks," *Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW04)*, pp. 427-433, 2004.